

# 基于分水岭算法的细胞图像分割及其改进

杨博文<sup>1</sup>、汲浩冉<sup>2</sup>、张奕驰<sup>3</sup>

(1. 北京师范大学 人工智能学院 计算机科学与技术专业, 202311081015  
\*作者简介: 杨博文, 北京师范大学人工智能学院计算机科学与技术专业本科在读, 学号 202311081015。  
汲浩冉, 北京师范大学人工智能学院计算机科学与技术专业本科在读, 学号 20231108246。  
张奕驰, 北京师范大学人工智能学院计算机科学与技术(公费师范)专业本科在读, 学号 202311081002。

摘 要: 摘要

关键词: 多模式智能系统; Arduino; 串口通信; 学习辅助系统; 传感器系统

## 1 项目背景与意义

应当包括: 所选问题讲解, 背景, 意义

## 2 实验数据说明

应当包括: 数据集说明, 数据集使用说明(只用 stage1\_train, 共 670 张图片, U-Net 部分如何分割), 四张典型图的选择说明(分别是: 分割、车轮、光、甜甜圈), 包括原图、GT-Mask 图(答案)

## 3 模型原理

### 3.1 Otsu 算法讲解

Otsu 算法是一类基于图像灰度直方图的自动阈值分割方法。其基本思想是通过最大化类间方差来确定最佳阈值, 从而将图像分割为前景和背景两部分。Otsu 算法的步骤如下:

1. 计算输入图像的归一化直方图。使用  $p_i$  表示该直方图的各个分量。
2. 用下式, 对  $k=0$  到  $L-1$  ( $L$  为灰度级数), 计算累计和  $p_1(k)$ 。

$$p_1(k) = \sum_{i=0}^k p_i \quad (1)$$

3. 用下式, 计算累计均值  $m(k)$ 。

$$m(k) = \sum_{i=0}^k i \cdot p_i \quad (2)$$

4. 计算全局灰度均值  $m_G$ 。

$$m_G = \sum_{i=0}^{L-1} i \cdot p_i \quad (3)$$

5. 对于  $k=0$  到  $L-1$ , 计算类间方差  $\sigma_B^2(k)$ 。

$$\sigma_B^2(k) = \frac{[m_G \cdot p_1(k) - m(k)]^2}{p_1(k) \cdot [1 - p_1(k)]} \quad (4)$$

6. 得到最大类间方差对应的  $k$  值, 即为最佳阈值  $k^*$ 。

### 3.2 分水岭算法讲解

分水岭算法是一种基于拓扑学的图像分割算法, 主要应用于将图像分割为多个具有连通性或相似性的区域, 尤其适用于从背景中提取近乎一致的物体。对于分水岭算法, 我们常常将灰度图像视为一幅地形图, 其中灰度值则表示地形的高度: 灰度值较小的区域我们认为是“低洼地带”, 而灰度值较大的区域则对应“山脊”。我们从“低洼地带”开始注水(即灰度值较小的区域), 水位会不断地上升, 当水位上升至一定高度时, 不同的水源会汇合, 此时, 我们需要在汇合处建立“堤坝”作为两股水源的分界线。随着水位的继续上升, 更多的“堤坝”会被建立起来, 直到整个图像都被水淹没为止。最终, 这些“堤坝”就形成了图像的分割边界。

### 3.3 U-Net 模型讲解

U-Net 是一种针对生物医学图像分割任务的卷积神经网络架构, 由 Olaf Ronneberger 等人于 2015 年提出<sup>[1]</sup>。如图 ?? 所示, 该模型由两部分组成: 一个收缩路径用于捕获上下文信息, 和一个对称的扩展路径用于精确的局部化。这种结构使

得网络能够在有限的训练样本下，提供高精度的分割结果。U-Net 的关键设计思想是通过上采样操作恢复高分辨率特征，并与收缩路径中的相应特征图进行拼接，从而提高分割的精确度。

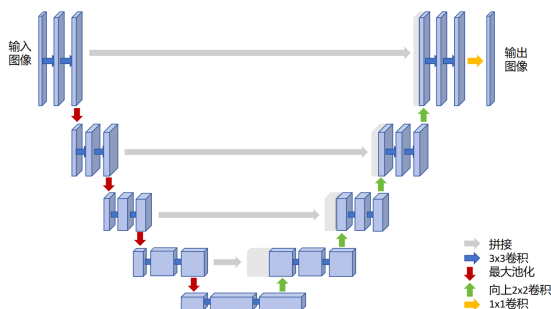


图 3.1 U-Net 模型原理示意图

Fig. 3.1 Schematic Diagram of the U-Net Model

U-Net 的网络结构没有全连接层，而是通过卷积层逐步提升图像的分辨率。在训练过程中，U-Net 采用了数据增强技术，这使得网络能够更好地学习到图像中的变形特征，增强其鲁棒性和泛化能力。该架构被广泛应用于多种生物医学图像分割任务，优势在于即使在训练数据较少的情况下，依然能够提供高质量的分割效果，并且具有较快的处理速度。

## 4 项目主要工作与解决方法

### 4.1 实验总流程

本实验将以传统的阈值分割算法——Otsu 算法为起点，结合分水岭算法和深度学习中的 U-Net 模型，逐步优化细胞分割的效果。其中，分水岭算法将采用基于距离变化和基于局部极大值的注水点选取方法，分别观察细胞分割效果的提升情况，并尝试引入梯度进行进一步的效果优化。最后，我们将引入 U-Net 模型进行前后景分割，结合前述的分水岭算法，进一步提升细胞分割的准确性，在这一部分中，我们将分别采取形态学，固定/可变阈值以及梯度的手段来分别优化分水岭算法的效果。具体的实验流程如图 ?? 所示。

在本次实验中，我们将通过视觉与数值两种方式对各个方法的分割效果进行评估。视觉评估主要通过观察分割结果图像（主要观察是否识别出所有细胞、细胞粘连问题是否解决以及是否存在过分割问题），视觉效果将采用如图 ?? 的样本图进行效果观察；数值评估则通过计算验证集中所有图片细胞分割数量的相对误差（基准细胞分

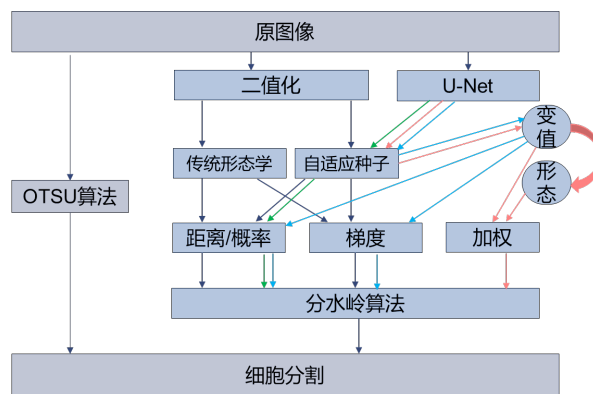


图 4.1 实验总流程图

Fig. 4.1 Schematic Diagram of the Experimental Flow

割数量由 kaggle 平台提供) 的平均值来评估，数值越小，则说明分割效果越好。

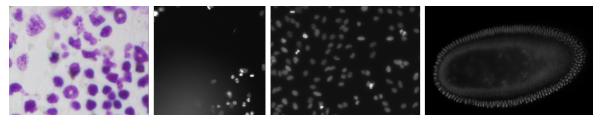


图 4.2 样本图

Fig. 4.2 Example of Sample Image

### 4.2 使用 Otsu 算法进行细胞分割

如前文所述，Otsu 算法是一种经典的基于图像灰度直方图的自动阈值分割方法。我们首先尝试使用 Otsu 算法对细胞图像进行二值化处理，以实现初步的细胞分割。实现步骤也较为简单，我们首先利用 Otsu 算法计算出最佳阈值（可调库），然后根据阈值进行二值化处理（即前景与背景的划分），最后对二值化结果进行连通域分析，统计细胞数量。分割结果如图 ?? 所示。

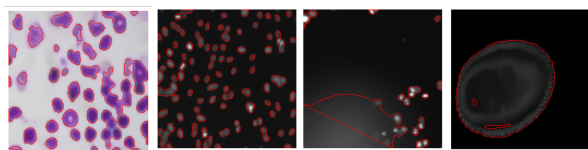


图 4.3 基于 Otsu 算法的阈值分割结果

Fig. 4.3 The Result of Otsu Algorithm-based Threshold Segmentation

首先，该算法细胞分割数量的平均相对误差为 0.2920，可以发现效果并不理想。我们再观察图 ??，可以发现该方法在处理细胞粘连问题上表现较差，许多粘连的细胞被错误地识别为单个细胞，导致分割数量偏少。并且，我们发现当噪声较大，或者细胞粘连十分严重时，Otsu 算法无法有效地分割细胞。

#### 4.3 使用基于距离变换的分水岭算法进行细胞分割

观察发现, 基于 Otsu 算法的细胞分割效果并不理想, 存在着细胞粘连无法分割等问题。而我们知道, 分水岭算法在处理图像分割问题上具有较好的效果, 尤其是在处理粘连物体的分割上表现突出。因此, 我们尝试引入分水岭算法来改进细胞分割效果。需要注意的是, 对于分水岭算法来说, 预处理过程尤其重要, 主要分为两个部分: 二值化过程——用于确定前景和背景; 注水点选取——选取细胞的核心区域。在这里, 我们主要考虑如何选取注水点, 二值化过程我们仍然采用 Otsu 算法进行处理。注水点的选取我们将采取基于距离变换和基于局部极大值两种方法, 分别观察其分割效果。

首先, 我们介绍基于距离变换的注水点选取方法。具体来说, 我们首先对二值化结果进行距离变换, 得到每个前景像素点到最近背景像素点的距离值。然后, 我们通过设定一个阈值——前景到背景最大的距离的某个比例, 来选取注水点。我们认为, 当距离值大于设定阈值时, 则像素点距离背景边界越远, 则越可能属于细胞的核心区域, 因此我们将这些点作为注水点。接着, 我们从这些注水点开始不断地向外膨胀, 直到遇到背景或两个膨胀区域相遇为止, 建立分水岭线, 完成细胞的分割。分割结果如图 ?? 所示。

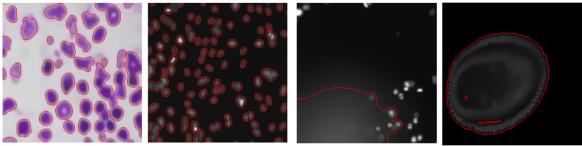


图 4.4 基于距离变换的分水岭结果

Fig. 4.4 The Result of Distance Transform-based Watershed Segmentation

该算法的细胞分割数量的平均相对误差为 0.2644, 相较于单纯的 Otsu 算法有了显著提升。观察图 ??, 可以发现该算法在处理细胞粘连问题上有了进一步的改善, 相较于单纯的 Otsu 算法, 可以对一些粘连细胞进行分割, 但是仍然存在大部分粘连细胞无法分割的问题。此外, 当噪声较大, 或者细胞粘连十分严重时, 同 Otsu 算法一样, 无法有效地分割细胞。

#### 4.4 使用基于局部极大值的分水岭算法进行细胞分割

接下来, 我们介绍基于局部极大值的注水点选取方法。由于基于距离变换的方法的距离阈值是基于全局信息设定的, 且是固定的。那么, 当设定的阈值过大时, 可能会出现某些距离背景过近的细胞核心区域无法被选取为注水点, 导致这些细胞无法被分割; 而当设定的阈值过小时, 则会出现粘连细胞的注水点过于接近从而识别为同一个连通区域, 无法进行粘连细胞的有效分割。为了解决上述问题, 我们引入了自适应的基于局部极大值的注水点选取方法。

具体来说, 我们首先对二值化结果进行距离变换。然后, 我们考虑每个前景像素点与其周围前景像素点的距离值进行比较, 若该像素点的距离值大于其所有周围像素点的距离值, 则认为该像素点为局部极大值点, 即细胞的核心区域。接着, 我们从这些注水点开始不断地向外膨胀, 直到遇到背景或两个膨胀区域相遇为止, 建立分水岭线, 完成细胞的分割。分割结果如图 ?? 所示。

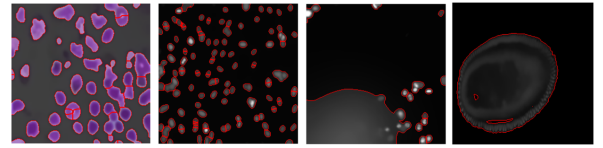


图 4.5 基于局部极大值的分水岭结果

Fig. 4.5 The Result of Local Maxima-based Watershed Segmentation

该算法的细胞分割数量的平均相对误差为 0.2318, 相较于基于距离变换的分水岭算法有了进一步提升。观察图 ??, 可以发现该算法在处理细胞粘连问题上表现更好, 能够有效地分割大部分粘连细胞。但是, 仍然存在小部分粘连细胞无法分割的问题。并且, 该算法有了新的问题——过分割问题, 会发现有些单个细胞被错误地分割成多个部分, 导致分割数量偏多。同时, 当噪声较大或粘连细胞过于严重时, 仍然无法有效地分割细胞。

#### 4.5 使用梯度优化分水岭算法进行细胞分割

基于教材的说法, 分水岭分割的主要应用之一是从背景中提取近乎一致的物体。由变化较小的灰度表征的区域有较小的梯度值。因此, 我们经常见到分水岭分割方法用于一幅图像的梯度, 而不是图像本身。为此, 我们尝试在前述的两种分水岭算法的基础上, 引入梯度信息来优化分割

效果。

我们首先对梯度图像进行归一化处理，然后将归一化后的梯度图像与二值化图像进行加权融合，得到新的图像。接着，我们分别在基于距离变换和基于局部极大值的分水岭算法中，使用该融合图像进行注水点选取和分割。分割结果如图 ?? 和图 ?? 所示。

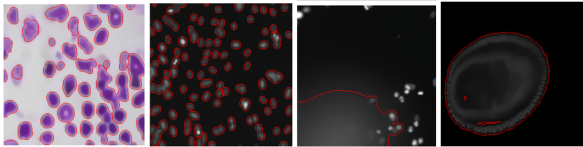


图 4.6 基于距离变换的梯度优化分水岭结果  
Fig. 4.6 The Result of Gradient-optimized Distance Transform-based Watershed Segmentation

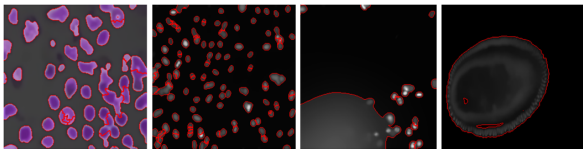


图 4.7 基于局部极大值的梯度优化分水岭结果  
Fig. 4.7 The Result of Gradient-optimized Local Maxima-based Watershed Segmentation

基于距离变换的梯度优化分水岭结果和基于局部极大值的梯度优化分水岭结果的细胞分割数量的平均相对误差分别为 0.2644 和 0.2318。观察图 ?? 和图 ??，可以发现引入梯度信息后，算法并没有进一步得到改善。

#### 4.6 使用 U-Net 进行前后景分割

如上文所言，我们在使用简单二值化方法时发现，二值化方法具有阈值不好确定、对于特殊图像效果极差的问题，所以我们考虑引入常用于医学图像分割的 U-Net 进行前后景分割。

由于这里我们只是引入 U-Net 卷积神经网络作为一个小小的步骤，而不是整个流程的核心，所以也不算是违背了最好不要使用神经网络简单训练一个模型的要求。

具体来讲，对于训练集，我们以其原图像和结果细胞图（原数据集中每个细胞一张图，这里将其合并为一张图）作为样本和标签，使 U-Net 具有能够区分每个点是前景（细胞）的概率，然后根据概率给定阈值，高于阈值的则认为其为细胞的组成部分，反之则认为其为背景。

首先，我们测试仅改变二值化方法（简单二值化、U-Net 二值化），即 U-Net+ 自适应分水岭

方法 + 对原图使用分水岭。依然考虑我们四张典型图片，分割结果如图 ??。

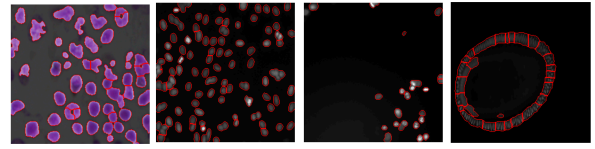


图 4.8 改进的基于 U-Net 做前景标记的自适应分水岭结果

Fig. 4.8 The Result of Improved Adaptive Watershed with U-Net Foreground Markers

首先，从分割细胞数量平均相对误差来看，其结果相较于改进的自适应分水岭（基于原图）结果，在其他变量均不变的情况下，从 0.2318 提升降低到了 0.1746。这是一个十分显著的提升。观察结果图，我们可以看到，其分割结果好了许多，尤其是第三、四图。第三图的左下角部分的光线没有被识别为细胞进行分割。第四图没有被完全整体识别为一个细胞，其环形结构被识别，部分边界被分开。第一、二图的结果也比之前的要更好，一部分连结细胞被分开。但是仍然存一些欠分割问题。

#### 4.7 使用变值方法基于细胞大小进行注水点选取

如上文所言，我们的自适应分水岭算法是根据极大值点确定注水点的，其中有一定的要求，比如说  $\min\_distance$  代表如果两个极大值点的差距小于  $\min\_distance$  的话就不将其视为两个。那么这里我们发现，这对于不同细胞的适应能力是较差的。有的细胞比较大，那个数值就应当相应的提高，反之则应当相应变小。所以接下来我们使用变值方法决定  $\min\_distance$ 。

这我们定义， $\min\_distance$  是 0.8 倍的最大距离（后续 0.8 会进行修改，在最后进行消融实验）。实验结果如图 ??。

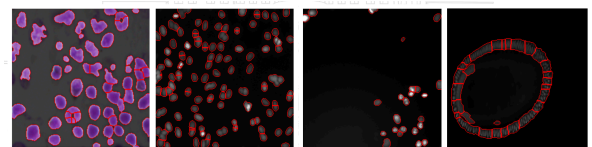


图 4.9 改进的基于 U-Net 做前景标记的使用变阈值的自适应分水岭结果

Fig. 4.9 The Result of U-Net-based Adaptive Watershed with Adaptive Thresholds

从分割细胞数量平均相对误差来看，其达到



0.1721, 相比前文有所增长但幅度不大。从结果图来看, 其对第二图的结果进行了尤其的改变, 分割更多了, 解决了一定的欠分割问题。与此同时, 其导致了过分割与欠分割并存的问题。

#### 4.8 使用梯度作为分水岭图像

我们再来尝试梯度图像。在本章前半部分, 在基于二值化的分割方法之中, 我们发现梯度对于结果的影响几乎没有。这是真的吗? 如果是这样的话为什么教材中会写一般使用梯度图像进行分水岭呢? 于是在我们使用 U-Net 做前景标记之后, 我们决定使用梯度图像看一下效果。结果如图 ??。

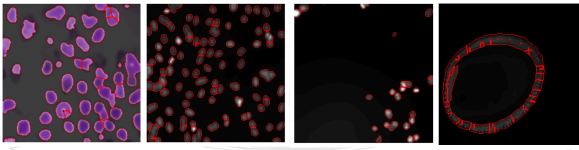


图 4.10 改进的基于 U-Net 做前景标记的使用变阈值的使用梯度图像的自适应分水岭结果

Fig. 4.10 The Result of U-Net-based Adaptive Watershed with Adaptive Thresholds and Gradient Images

从分割细胞数量平均相对误差来看, 其达到 0.1346, 相较前文有较大的提升。从结果图上来看, 其更清晰的关注到了边界, 开始关注内部的线条。但是, 其对于概率图变色过于敏感可能导致结果不够好, 如第四图出现从中间直接连到边界的不完全分割情况。

#### 4.9 使用梯度作为分水岭图像

既然梯度单独不行, 那么我们同时考虑梯度和原始概率图, 将二者加权以期更好的结果。结果如图 ??。

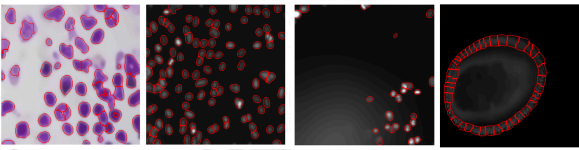


图 4.11 改进的基于 U-Net 做前景标记的使用变阈值的使用梯度-概率加权图像的自适应分水岭结果

Fig. 4.11 The Result of U-Net-based Adaptive Watershed Using Gradient-Probability Weighted Images

从分割细胞数量平均相对误差来看, 其达到 0.1323, 相较前文又有一定的提升。从结果图上来看, 其改进了从中间直接连到边界的不完全分割情况, 分割结果越来越好。但是, 注意第一图中下部分, 出现“甜甜圈”现象。即, 对于细胞

和细胞核, 呈现从细胞核到细胞边缘被切成多份的情况。

#### 4.10 使用形态学方法改善“甜甜圈”问题

我们引入形态学内容来改进“甜甜圈”现象。当一个分割结果形状接近圆、椭圆且离心率较小的时候, 我们认为这已经是一个分割好的细胞了不再需要进行分割。其结果如图 ??。

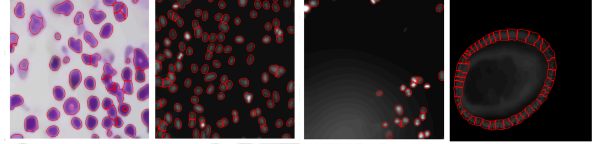


图 4.12 改进的基于 U-Net 做前景标记的加入形态学判断的使用变阈值的使用梯度-概率加权图像的自适应分水岭结果

Fig. 4.12 The Result of U-Net-based Adaptive Watershed with Morphology and Weighted Images

从分割细胞数量平均相对误差来看, 其变差了一点, 变为 0.1334。但是从结果图上来看, 其变好了一些, 第一图的“甜甜圈”现象被正确解决。详细分析见本文第 5 章。

### 5 实验结果展示及分析

主要内容应当包括: 总实验结果图

在本章中, 我们将聚焦于改进的基于 U-Net 做前景标记的加入形态学判断的使用变阈值的使用梯度-概率加权图像的自适应分水岭结果, 进行一定的分析, 并且给出后续可能的改进方向。

### 6 消融实验

#### 6.1 传统分水岭算法参数选取

在传统分水岭算法中, 参数的选取对于分割效果有着重要影响。我们在前面说到, 当距离阈值的选取不合适时, 可能会导致某些细胞无法被识别, 或者出现细胞粘连无法分割的问题。而我们的距离阈值选取主要依赖于距离变换结果的最大值的某个比例。因此, 我们尝试调整该比例参数, 以观察其对分割效果的影响。在本次实验中, 我们选取了 0.1、0.2、0.3、0.4、0.5、0.6 六个不同的比例参数, 分别进行细胞分割实验, 并计算其细胞分割数量的平均相对误差。实验结果如表 ??所示。

可以发现, 当距离阈值比例设定为 0.2 时, 分割效果最佳, 平均相对误差最低, 为 0.2644。因

表 1 传统分水岭算法参数选取结果

Table 1 The Result of Traditional Watershed Algorithm

Parameter Selection

| 距离阈值比例 | 平均相对误差 |
|--------|--------|
| 0.1    | 0.2742 |
| 0.2    | 0.2644 |
| 0.3    | 0.2691 |
| 0.4    | 0.2778 |
| 0.5    | 0.3490 |
| 0.6    | 0.4609 |

此, 在后续的传统分水岭算法实验中, 我们均采用该参数进行细胞分割。

## 6.2 U-Net 算法参数选取

在引入 U-Net 的算法中, 我们使用了基于局部极大值的分水岭算法进行了注水点的选取。而在这个过程中, 我们同样设定一个阈值——两个注水点之间的最小距离, 以避免选取过于接近的注水点, 导致过分割问题。同样, 这个阈值是依赖于局部区域的距离最大值乘以一个比例系数来确定的。在实验过程中, 我们尝试了 0.5、0.6、0.7、0.8、0.9、0.95 六个不同的比例系数, 分别进行细胞分割实验, 并计算其细胞分割数量的平均相对误差。实验结果如表 ?? 所示。

表 2 U-Net 算法参数选取结果

Table 2 The Result of U-Net Algorithm Parameter Selection

| 最小距离比例 | 平均相对误差 |
|--------|--------|
| 0.5    | 0.1340 |
| 0.6    | 0.1335 |
| 0.7    | 0.1335 |
| 0.8    | 0.1337 |
| 0.9    | 0.1334 |
| 0.95   | 0.1335 |

可以发现, 当最小距离比例设定为 0.9 时, 分割效果最佳, 平均相对误差最低, 为 0.1334。因此, 在后续的 U-Net 算法实验中, 我们均采用该参数进行细胞分割。

## 7 总结

应当包括: 我们干了什么, 有什么提升, 算法的适用条件 (基于 2018dsb 的优化, 没有在其

他数据集上进行测试, 认为: 针对清晰、较为一致的更优质细胞数据集有极强的能力 (比如: 在这个数据集之中的比较好的照片就有比较好的结果, 图片到群里找, 要求分割基本完全正确), 对于原始图像质量较差的细胞图像具有一定的分割能力, 可以看到一些人眼看不到的细胞。如果我们能有这样一个初步结果, 然后再人工来个别判断, 会节省很大的时间 (这个在群里的聊天记录里面我发过)), 可以改进的方向 (二值化 +U-Net 的探索 (前面会对 mix 进行讨论, 这里简单说就可以了), U-Net 的 binary, 更好的参数选择 (我们没有对每一个参数做调整, 许多基于经验等等))。

## 参考文献 (References)

- [1] Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. MICCAI 2015, 234-241. Springer, Cham. DOI: 10.1007/978-3-319-24574-4\_28
- [2] BADAMASI Y A. The working principle of an Arduino[C]//2014 11th International Conference on Electronics, Computer and Computation (ICECCO). Abuja, Nigeria: IEEE, 2014: 1 - 4. doi:10.1109/ICECCO.2014.6997578.
- [3] ASADI F. Serial Communication[M]//Essentials of Arduino™ Boards Programming: Step-by-Step Guide to Master Arduino Boards Hardware and Software. Berkeley, CA: Apress, 2023: 179-199.
- [4] EVANS B. Serial and I2C[M]//Beginning Arduino Programming. Berkeley, CA: Apress, 2011: 175 - 200.
- [5] AKINWOLE O O, OLADIMEJI T T. Design and implementation of Arduino microcontroller based automatic lighting control with I2C LCD display[J]. Journal of Electrical and Electronic Systems, 2018, 7(258): 2332-0796.1000258.
- [6] LEOTS. Arduino-PC-Monitor[EB/OL]. <https://github.com/leots/Arduino-PC-Monitor>, 2023-12-20.
- [7] GGARCIA. DIY PC Stats Monitor[EB/OL]. <https://www.hackster.io/ggarciacompanys/diy-pc-stats-monitor-2eb973>, 2023-10-15.

- [8] ANSH2919. Serial communication between Python and Arduino[EB/OL]. <https://projecthub.arduino.cc/ansh2919/serial-communication-between-python-and-arduino-663756>, 2022-08-02.
- [9] DeepSeek. DeepSeek API Documentation[EB/OL]. <https://api-docs.deepseek.com/zh-cn/>, 2024-01-10.
- [10] DOS SANTOS D A, SILVA H P S, DOS ANJOS COSTA D. Desenvolvimento de um Braço Robótico controlado por Joystick[C]//Escola Regional de Computação Bahia, Alagoas e Sergipe (ERBASE). SBC, 2023: 82-91.
- [11] PREMENDRAN K, KULATUNGA U, JAYASINGHE A et al. Assistant Zone - Homeschooling Assistance System based on Natural Language Processing[C]//2022 4th International Conference on Advancements in Computing (ICAC). Colombo, Sri Lanka: IEEE, 2022: 13 - 17. doi:10.1109/ICAC57685.2022.10025201.